ArcGIS GeoEvent Server Add-ons and Custom Components: Version Compatibility and How to Compile

This document provides information on version compatibility of add-ons and custom components for ArcGIS GeoEvent Server and how to compile the Java Archive (JAR) files for them from GitHub repositories.

Add-on and custom component version compatibility

GeoEvent Server add-ons and custom components can be compiled against any version of the ArcGIS GeoEvent Server SDK. Note that the version of the GeoEvent Server SDK an add-on or custom component is compiled against does not need to match the version of GeoEvent Server you have deployed in your organization.

Register add-ons and custom components

All add-ons and custom components in GeoEvent Server are uniquely registered using a Uniform Resource Identifier (URI) that consists of *<domain> | <symbolic-name> | <version>*. A description of each follows:

- *domain*—The general group the component belongs to (for example, *com.esri.geoevent.adapter*).
- *symbolic-name*—The unique name for the component (for example, *field-calculator*).
- *version*—The version of the GeoEvent Server SDK used to compile and build the component (for example, *11.1.0*).

Add-on and custom component versions do not need to match the version of GeoEvent Server

The version of an add-on or custom component that you install in GeoEvent Server does not need to match the version of ArcGIS GeoEvent Server you are using. In general, add-ons and custom components compiled with a previous version of the GeoEvent Server SDK should work with later versions of GeoEvent Server. Because of this, and the reasons outlined below, it is common for the version of an add-on or custom component to not match the current version of GeoEvent Server that is available or that you have deployed in your organization. In addition, add-ons and custom components typically do not need to be recompiled against newer versions of the SDK when you upgrade GeoEvent Server in your organization.

Occasionally, there may be changes to policies and libraries in a new version of GeoEvent Server which can cause your existing add-ons and custom components not to work properly or install in GeoEvent Server. To address this, you will need to recompile your add-ons and custom components with a newer version of the GeoEvent Server SDK.

Existing add-ons and custom components can be used when upgrading ArcGIS GeoEvent Server

When you upgrade GeoEvent Server to a new version, or import a configuration from a previous version, GeoEvent Server will automatically update the URI of all the out-of-the-box components so their versions are consistent with the version you are upgrading to. However, upgrading GeoEvent Server to a new version will not update the version of the URI for any add-ons or custom components you have installed. If you recompile your add-ons and custom components against a newer version of the SDK, the URIs would no longer match the URIs in the existing GeoEvent Server configuration. All elements you have that use the add-ons and custom component will fail to load and errors will exist in your configuration. For this reason, it is not recommended you recompile add-ons and custom components to match the version of an upgraded GeoEvent Server. Instead, it is recommended you compile your add-ons and custom components against the earliest version of the GeoEvent Server SDK you plan to deploy against.

Esri managed add-ons have a release number

For the reasons above, add-ons released and supported by Esri in the GeoEvent Server Gallery are based on a release number instead of the version of the GeoEvent Server SDK they are compiled against. The release number does not impact the configuration of GeoEvent Server (it is not part of the add-on's URI) and can be incremented every time a change is made to the add-on.

For Esri supported add-ons, a new release is only necessary when:

- A bug is found.

- Enhancements with new capabilities are required.

- The current release is not compatible with new versions of GeoEvent Server.

Compile JAR files for add-ons and custom components from GitHub repositories

To deploy an add-on or a custom component to GeoEvent Server, a Java Archive (JAR) file is needed. However, some of the GeoEvent Server GitHub repositories for add-ons and custom components do not provide a JAR file. When a JAR file is not available, you can compile it yourself from a GitHub repository.

Verify system requirements

To get started, you must have the following software installed on your system:

- AdoptOpenJDK:
  - Version 8 if you are building for ArcGIS GeoEvent Server 10.8.1 and below.
  - Version 11 if you are building for ArcGIS GeoEvent Server 10.9 and above.

- Apache Maven version 3.6.3 (or later).

To check if you have the right versions of the above software installed, run the following commands in a command line. Note that the specific command to check the versions may vary based on the actual version you have installed.

```
javac -version
```

```
mvn -version
```

Clone the GitHub repository

Next, you will clone the GitHub repository to your local system. You can do this using a variety of methods including downloading the zip file directly from the repository on the GitHub site or using a GIT client. Follow the steps below to use the GIT command line utility:

1. Copy the URL to the target GitHub repository for the add-on JAR file you want to create.
2. On your system, open a command line as an administrator.
3. Create a temporary folder and navigate into it.
4. Run the command below (as an administrator) to clone the target GitHub repository. This downloads the entire repository into the empty temporary directory.

   ```
   git clone https://github.com/Esri/add-on
   ```

   Note: *Replace the URL in the command above (https://github.com/Esri/add-on) with the URL you copied in Step 1.*

5. Run the command below to navigate into the root of the downloaded GitHub repository project folder.

   ```
   cd add-on-folder
   ```

   Note: *Replace the folder name in the command (add-on-folder) with the folder name you created in Step 3 and cloned the repository to in Step 4.*

Build the project

You are now ready to build the project.

1. Run the command below to instruct Maven to compile the repository into a JAR file.

   ```
   mvn clean package
   ```

   Note: *Clean will delete any previously compiled components on the system and package compiles the source code and builds the* JAR *file.*

2. After the command runs, a *target* subfolder is created in the project.
   a. Some projects may have multiple modules, resulting in multiple JAR file artifacts.
   b. Each module subfolder will contain a *target* subfolder.

3. Navigate into the *target* subfolder and verify JAR files exist with the project’s name and the GeoEvent Server SDK version it was compiled against.
4. Install the JAR files in GeoEvent Server by following the information in Install and upgrade add-ons manually.